NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY POLITEHNICA BUCHAREST

Doctoral School of Electronics, Telecommunications and Information Technology

Ph.D. THESIS SUMMARY

Ionuț-Dorinel FÎCIU

ALGORITMI ADAPTIVI EFICIENȚI BAZAȚI PE OPTIMIZAREA ÎN SENSUL CELOR MAI MICI PĂTRATE PENTRU IDENTIFICAREA SISTEMELOR

IMPROVED RECURSIVE LEAST-SQUARES ADAPTIVE ALGORITHMS FOR SYSTEM IDENTIFICATION PROBLEMS

Prof. Dr. Ing. Constantin PALEOLOGU — PhD Supervisor
POLITEHNICA Bucharest

BUCHAREST 2025

Table of contents

Chapter 1

Introduction

This thesis explores advancements in adaptive filtering within the field of signal processing, driven by the need for efficient data analysis in increasingly complex technological systems. With the rise in data volume and the necessity for robust techniques in non-stationary environments, adaptive filtering has become crucial. The study focuses on improving recursive least squares (RLS) adaptive algorithms for system identification, a process vital for creating accurate mathematical models of dynamic systems. These models have significant applications in control systems, predictive maintenance, and adaptive signal processing, especially where noise and non-stationarity represent challenges. This chapter introduces key concepts and establishes the foundational principles that support the research presented in this work.

1.1 Presentation of the field of the doctoral thesis

Signal processing is a multidisciplinary field focused on analyzing, manipulating, and interpreting signals that represent information about physical systems. The main goal is to extract meaningful information while minimizing noise and distortions. Filtering, a key concept, modifies signals to achieve desired outputs, with linear filters like finite impulse response (FIR) and infinite impulse response (IIR) being extensively studied.

Adaptive filtering allows filters to dynamically adjust parameters in response to changing data, making it valuable in environments with variable signal characteristics. The RLS algorithm is a widely used adaptive filtering technique known for rapid convergence and efficient computation, making it suitable for real-time applications such as telecommunications and audio processing. However, challenges like sensitivity to noise and computational complexity persist, especially in system identification.

The ongoing research in this field aims to address these limitations by developing improved algorithms that enhance robustness against noise, reduce computational complexity, and improve convergence rates. Such advancements contribute to effective solutions in applications that depend on accurate and efficient signal processing.

1.2 Scope of the doctoral thesis

This doctoral thesis aims to enhance RLS adaptive algorithms in signal processing and system identification through three key objectives. First, it integrates tensor decomposition with RLS algorithms to manage complex data structures and improve system identification efficiency. Second, it advances RLS algorithms by implementing a data-reuse approach, enhancing convergence rates and addressing computational challenges. Third, it applies RLS algorithms to stereophonic acoustic echo cancellation (SAEC), improving impulse response estimation and managing correlations in audio systems. Collectively, these objectives seek to develop innovative RLS algorithms that enhance performance and offer new perspectives in adaptive filtering and signal processing.

1.3 Content of the doctoral thesis

This doctoral thesis is structured into four main chapters, each addressing critical aspects of RLS adaptive algorithms in signal processing. Chapter 2 establishes the groundwork by examining the principles of adaptive filtering, focusing on key performance metrics and comparing configurations and categories of adaptive algorithms. It also discusses a specific algorithm and its enhanced version, highlighting their distinct benefits.

Chapter 3 delves into the integration of tensor decomposition with RLS algorithms for system identification, showcasing how it handles complex data structures and reformulates tasks into manageable problems, thus reducing computational demands. Various algorithms are assessed for their effectiveness in dynamic environments.

Chapter 4 presents a data-reuse (DR) strategy to refine RLS algorithms, aiming to boost convergence rates and filter performance by enabling multiple updates on the same input and reference signals. The chapter evaluates different variations of this strategy across diverse scenarios.

Chapter 5 explores the application of RLS algorithms in the context of SAEC, addressing challenges posed by multiple loudspeakers and microphones. It emphasizes a widely linear model and details adaptive techniques that enhance tracking and performance in complex acoustic settings.

Finally, Chapter 6 summarizes the research findings, highlights the original contributions to the field by listing published papers, and discusses future research directions.

Together, these chapters aim to advance the understanding and application of RLS adaptive algorithms, providing novel insights into adaptive filtering and signal processing.

Chapter 2

Introduction to adaptive filtering theory

This chapter delves into adaptive filtering principles, key performance metrics, and applications like echo cancellation and interference suppression. It examines adaptive system configurations and compares two main categories of adaptive algorithms, discussing their pros and cons. Additionally, it explores the RLS algorithm and its enhanced version with Dichotomous Coordinate Descent (DCD) iterations, focusing on their unique features, performance benefits, and challenges.

2.1 The principle of adaptive filtering

An adaptive filter is a digital filter that dynamically adjusts its parameters to optimize performance using a recursive algorithm [1]. It is beneficial in environments with non-stationary or unpredictable signals. The filter's parameters are controlled by an adaptive algorithm, determining continuously optimal coefficients based on a specified criterion.

The system involves two primary inputs: the filter input signal, $x(n)$, and the desired response, $d(n)$. The output signal $y(n)$ is designed to approximate $d(n)$, with the error signal defined accordingly:

$$e(n) = d(n) - y(n) \tag{2.1}$$

The error signal $e(n)$ is essential for recursively updating filter coefficients by minimizing a cost function, such as mean squared error or ℓ_2 norm. Adaptive algorithms are selected based on robustness, complexity, misadjustment, tracking ability, structure, and convergence speed.

2.2 Adaptive system configurations

Adaptive filtering is vital in telecommunications for tasks like channel equalization, interference suppression, and echo cancellation, maintaining signal integrity. It is also important in AI and deep learning for real-time learning and adaptation. Various configurations of adaptive systems address specific challenges.

2.2.1 System identification

This configuration models an unknown system by adjusting the adaptive filter to match the system's output, commonly used for echo cancellation.

2.2.2 Inverse modeling

In this setup, the adaptive filter approximates the inverse transfer function of an unknown system, useful in audio engineering to correct distortions.

2.2.3 Prediction

This configuration estimates future signal values by comparing the adaptive filter's output to current samples, applied in financial forecasting.

2.2.4 Interference suppression

This setup reduces noise by using a reference signal correlated with the disturbance, improving signal clarity in telecommunications.

2.3 Main adaptive algorithms

Adaptive filtering has evolved significantly since the 1940s, with two primary families of algorithms emerging: those based on MSE optimization and those based on least squares (LS) optimization. The choice between these depends on application needs.

2.3.1 Mean square error optimization algorithms

These algorithms use the mean square error as a cost function and are exemplified by the Least Mean Square (LMS) algorithm. The LMS algorithm minimizes the MSE, making it robust and efficient for applications with limited computational resources. Its update rule is given by:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{x}(n) e^*(n), \tag{2.10}$$

where $e(n)$ is the error signal. Stability is maintained by controlling the step size μ, with the stability interval:

$$0 < \mu < \frac{2}{\lambda_{\max}}, \tag{2.11}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the input signal's autocorrelation matrix. The Normalized LMS (NLMS) variant adjusts the step size based on input power, enhancing robustness.

2.3.2 Least squares optimization algorithms

These algorithms minimize the weighted sum of squared errors, offering faster convergence but with higher complexity. The RLS algorithm is a key example, providing superior convergence speed, being ideal for dynamic environments. The normal equation for LS criterion is:

$$\mathbf{R}(n)\widehat{\mathbf{h}}(n) = \mathbf{p}(n), \tag{2.19}$$

where $\mathbf{R}(n)$ is the autocorrelation matrix and $\mathbf{p}(n)$ is the correlation vector. The RLS algorithm recursively updates these values to compute optimal filter coefficients, balancing complexity and adaptability.

2.4 Recursive least squares algorithm

The RLS algorithm minimizes an exponentially weighted sum of squared errors using a cost function that includes a forgetting factor, λ. This factor affects memory and responsiveness: smaller values enable faster adaptation, while larger values provide beneficial long-term memory in stationary environments. The cost function is updated recursively, allowing efficient computation of the autocorrelation matrix and correlation vector, which facilitates real-time estimation of filter coefficients.

2.4.1 Recursive relations

The RLS algorithm employs recursive relations for cost function updates, maintaining low computational complexity and enabling timely updates of filter coefficients. It initializes with specific values for filter coefficients and the inverse covariance matrix.

2.4.2 Orthogonality relations

The algorithm adheres to orthogonality conditions, minimizing the apriori error for optimal coefficients and confirming the RLS approach's effectiveness in dynamic settings.

2.5 The exponential weighted RLS-DCD algorithm

The RLS-DCD algorithm builds on the RLS framework to reduce computational complexity while preserving performance, focusing on the residual vector for efficient filter coefficient updates.

It features recursive updates for the correlation matrix and cross-correlation vector, essential for adapting the estimated impulse response. The RLS-DCD method shows greater stability and efficiency compared to classical RLS approaches, proving effective in real-time applications.

Chapter 3

Recursive least-squares adaptive algorithms based on tensor decomposition

This chapter investigates the enhancement of recursive least-squares (RLS) adaptive algorithms through tensor decomposition to improve system identification. By efficiently managing complex data structures, these methods reduce computational complexity and enhance performance in fields like communications and signal processing.

The research demonstrates that tensor decomposition allows for breaking down system identification tasks into smaller, manageable problems, reducing computational demands while maintaining or improving performance. These algorithms show promise for real-time applications, offering robustness and efficiency.

The techniques and findings discussed in this chapter have been published in [2–7].

3.1 Multilinear forms system model

In system identification, tensor decomposition efficiently manages complex data structures. This is applied in a multiple-input/single-output (MISO) system, where the output signal $y(n)$ at discrete-time index n is defined as the aggregation of input signals using the tensor $\mathcal{X}(n) \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_N}$:

$$y(n) = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \cdots \sum_{l_N=1}^{L_N} x_{l_1 l_2 \ldots l_N}(n) h_{1,l_1} h_{2,l_2} \cdots h_{N,l_N}. \tag{3.2}$$

The desired signal $d(n)$ incorporates uncorrelated noise $w(n)$:

$$d(n) = \mathbf{g}^T \mathbf{x}(n) + w(n). \tag{3.12}$$

3.2 The NLMS-T algorithm

The NLMS-T algorithm addresses the system identification problem via adaptive filtering, where the error signal is defined as:

$$e(n) = d(n) - \widehat{\mathbf{g}}^T(n-1)\mathbf{x}(n). \tag{3.15}$$

Initialization of the filter coefficients is crucial, defined as:

$$\widehat{\mathbf{h}}_1(0) = \begin{bmatrix} 1 \\ \mathbf{0}_{L_1-1} \end{bmatrix}, \quad \widehat{\mathbf{h}}_j(0) = \frac{1}{L_j}\mathbf{1}_{L_j}, \quad j = 2, 3, \ldots, N, \tag{3.20}$$

This gives reasonable starting estimates for channel responses. Each $\widehat{\mathbf{h}}_i(n)$, for $i = 1, 2, ..., N$, is updated using an NLMS algorithm, resulting in the global estimate:

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{h}}_N(n) \otimes \cdots \otimes \widehat{\mathbf{h}}_1(n). \tag{3.21}$$

The Kronecker product facilitates effective combination of individual filter contributions, enhancing performance in adaptive filtering through the NLMS-T algorithm.

3.3 The RLS-T algorithm

RLS-based algorithms converge faster than the LMS family but require more computation. The tensor-based approach improves RLS efficiency by adapting the LS error criterion [1] to a tensorial context using relations from (3.12) and (3.15). The RLS-T algorithm employs cost functions in N forms, each corresponding to an impulse response, and uses multilinear optimization [8] by fixing $N-1$ components while optimizing one, leading to iteratively solved normal equations.

Key steps involve updating filters using Kalman gain vectors derived from the inverse of correlation matrices, updated via the matrix inversion lemma [1]. Initialization follows (3.20), and the global impulse response is determined similarly to (3.21). The tensor-based approach is computationally efficient, with complexity proportional to $\sum_{i=1}^{N} O(L_i^2)$, compared to the conventional RLS algorithm's $O(L^2)$, with $L = \prod_{i=1}^{N} L_i$. The RLS-T algorithm offers superior convergence and tracking, particularly with longer filters.

3.4 The RLS-NLMS-T algorithm

This section presents a solution that utilizes the NLMS-T and RLS-T algorithms (described in Sections 3.2 and 3.3) for individual filters introduced in [2, 3]. This integration of adaptive filters within multilinear forms achieves a better balance between convergence characteristics and computational complexity of the overall approach.

The developed tensor-based algorithms extend filters for bilinear forms, where performance relies on the longest filter. For multilinear forms, a combination of adaptive filters is proposed: using RLS-T for the first M filters and NLMS-T for the remaining ones. This method, referred to as RLS$_M$-NLMS$_{N\text{-}M}$-T, leverages RLS-T's fast convergence and NLMS-T's lower complexity. Simulations are anticipated to demonstrate significant performance improvements, even with the configuration using RLS for the first filter. Additionally, incorporating DCD iterations within the RLS-T algorithm may enhance efficiency.

Table 3.3 The RLS$_M$-NLMS$_{N\text{-}M}$-T algorithm

Step	**Actions**
Init.	Set $\widehat{\mathbf{h}}_i(0)$, based on (3.20) $\mathbf{R}_m^{-1}(0) = \delta_m^{-1}\mathbf{I}_{L_i},\ \delta_m > 0,\ m = 1,2,\ldots,M,\ M < N$ $\lambda_m = 1 - \frac{1}{K_m L_m},\ K_m \geq 1,\ m = 1,2,\ldots,M$ $0 < \alpha_p < 1,\ \xi_p > 0,\ p = M+1, M+2,\ldots,N$
For $n = 1,2,\ldots,$ number of iterations:	
1	Compute $\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$
2	$e_{\widehat{\mathbf{h}}_i}(n) = d(n) - \widehat{\mathbf{h}}_i^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$
3	For $m = 1,2,\ldots,M$: $\mathbf{k}_m(n) = \frac{\mathbf{R}_m^{-1}(n-1)\mathbf{x}_{\widehat{\mathbf{h}}_m}(n)}{\lambda_m + \mathbf{x}_{\widehat{\mathbf{h}}_m}^T(n)\mathbf{R}_m^{-1}(n-1)\mathbf{x}_{\widehat{\mathbf{h}}_m}(n)}$ $\widehat{\mathbf{h}}_m(n) = \widehat{\mathbf{h}}_m(n-1) + \mathbf{k}_m(n)e_{\widehat{\mathbf{h}}_m}(n)$ $\mathbf{R}_m^{-1}(n) = \frac{1}{\lambda_m}\left[\mathbf{I}_{L_m} - \mathbf{k}_m(n)\mathbf{x}_{\widehat{\mathbf{h}}_m}^T(n)\right]\mathbf{R}_m^{-1}(n-1)$
4	For $p = M+1, M+2,\ldots,N$: $\widehat{\mathbf{h}}_p(n) = \widehat{\mathbf{h}}_p(n-1) + \frac{\alpha_p \mathbf{x}_{\widehat{\mathbf{h}}_p}(n)e_{\widehat{\mathbf{h}}_p}(n)}{\xi_p + \mathbf{x}_{\widehat{\mathbf{h}}_p}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}_p}(n)}$
5	$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{h}}_N(n) \otimes \widehat{\mathbf{h}}_{N-1}(n) \otimes \cdots \otimes \widehat{\mathbf{h}}_1(n)$

3.5 The RLS-DCD-T algorithm

Table 3.4 The RLS-DCD-T algorithm for one channel with complexity

Step	**Actions**	**Complexity** '$\times$' & '+'
Init.	Set $\widehat{\mathbf{h}}_i(0) = \mathbf{0}_{L_i\times 1},\ \mathbf{r}_i(0) = \mathbf{0}_{L_i\times 1}$ $\mathbf{R}_i^{-1}(0) = \delta_i \mathbf{I}_{L_i},\ \delta_i > 0$ $\lambda_i = 1 - \frac{1}{K_i L_i},\ K_i \geq 1$	
For $n = 1,2,\ldots,$ number of iterations:		
1	Compute $\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$	$L + \sum_{i=1}^{N-1}\prod_{j=i}^{N} L_j$ & $L-1$
2	$\mathbf{R}_i^{(0)}(n) = \lambda_i \mathbf{R}_i^{(0)}(n-1) + \mathbf{x}_{\widehat{\mathbf{h}}_i}(n)\mathbf{x}_{\widehat{\mathbf{h}}_i}^{(0)}(n)$	$2L_i$ & L_i
3	$y_{\widehat{\mathbf{h}}_i}(n) = \widehat{\mathbf{h}}_i^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$	L_i & $L_i - 1$
4	$e_{\widehat{\mathbf{h}}_i}(n) = d(n) - y_{\widehat{\mathbf{h}}_i}(n)$	0 & 1
5	$\mathbf{p}_{0,i}(n) = \lambda_i \mathbf{r}_i(n-1) + e_{\widehat{\mathbf{h}}_i}(n)\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$	$2L_i$ & L_i
6	$\mathbf{R}_i(n)\Delta\mathbf{h}_i(n) = \mathbf{p}_{0,i}(n) \xrightarrow{\text{DCD}} \Delta\widehat{\mathbf{h}}_i(n), \mathbf{r}_i(n)$	0 & $N_{u,i}(2L_i+1) + M_b$
7	$\widehat{\mathbf{h}}_i(n) = \widehat{\mathbf{h}}_i(n-1) + \Delta\widehat{\mathbf{h}}_i(n)$	0 & L_i

The RLS-DCD-T algorithm combines DCD iterations with the RLS method from Section 2.5 to efficiently solve systems of equations [4, 5], enhancing computational efficiency and stability by leveraging previous results from each of the N filters to compute coefficient increments. It estimates filter coefficients using DCD methods, reducing effort in updating residual vectors. The steps and complexity are detailed in Table 3.4. DCD iterations lower arithmetic workload, enabling hardware implementation without multiplications, making it a more efficient alternative to traditional RLS methods for large filters and adaptive systems.

3.6 The VR-RLS-DCD-T algorithm

The VR-RLS-DCD-T algorithm [6, 7], presented in Table 3.6, improves RLS-DCD-T by dynamically adjusting δ_i to enhance low SNR performance and address ill-conditioning. It integrates real-time parameter estimation, aligning error with noise variance. It also outlines key steps like variance computation and SNR estimation, offering robustness with minimal cost, suitable for FPGA implementations.

Table 3.6 The VR-RLS-DCD-T algorithm

Step	Actions
Init.	Set $\widehat{\mathbf{h}}_i(0) = \mathbf{0}_{L_i \times 1}$; $\mathbf{r}_i(0) = \mathbf{0}_{L_i \times 1}$; $\mathbf{R}_i^{-1}(0) = \delta_i \mathbf{I}_{L_i}$, $\delta_i > 0$ $\lambda_i = 1 - \frac{1}{K_i L_i}$, $K_i \geq 1$; $\gamma_i = 1 - \frac{1}{\epsilon_i L_i}$, $\epsilon_i > 0$ $\widehat{\sigma}_{\widehat{y}}^2(0) = 0$; $\widehat{\sigma}_d^2(0) = 0$; $\widehat{\sigma}_x^2(0) = 0$
For $n = 1, 2, \ldots,$ number of iterations:	
1	Compute $\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$
2	$\mathbf{R}_i^{(0)}(n) = \lambda_i \mathbf{R}_i^{(0)}(n-1) + \mathbf{x}_{\widehat{\mathbf{h}}_i}(n)\mathbf{x}_{\widehat{\mathbf{h}}_i}^{(0)}(n)$
3	$\widehat{y}_{\widehat{\mathbf{h}}_i}(n) = \widehat{\mathbf{h}}_i^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}_i}(n)$
4	$e_{\widehat{\mathbf{h}}_i}(n) = d(n) - \widehat{y}_{\widehat{\mathbf{h}}_i}(n)$
5 a)	$\widehat{\sigma}_u^2(n) = \gamma_i \widehat{\sigma}_u^2(n-1) + (1-\gamma_i)u^2(n)$, for $u(n) \triangleq x(n), d(n), \widehat{y}(n)$
5 b)	Compute $\alpha_i(n)$
5 c)	$\widehat{\text{SNR}}(n) = \frac{\widehat{\sigma}_{\widehat{y}}^2(n)}{\left\|\widehat{\sigma}_d^2(n) - \widehat{\sigma}_{\widehat{y}}^2(n)\right\|}$
5 d)	$\delta_i(n) = \alpha_i(n)\frac{1+\sqrt{1+\widehat{\text{SNR}}(n)}}{\widehat{\text{SNR}}(n)}$
6	$\left[\mathbf{R}_i(n) + \delta_i(n)\mathbf{I}_{L_i}\right]\Delta\widehat{\mathbf{h}}_i(n) = \mathbf{p}_{0,i}(n) \xrightarrow{\text{DCD}} \Delta\widehat{\mathbf{h}}_i(n), \mathbf{r}_i(n)$
7	$\widehat{\mathbf{h}}_i(n) = \widehat{\mathbf{h}}_i(n-1) + \Delta\widehat{\mathbf{h}}_i(n)$

3.7 Simulation results

Simulations for MISO system identification use Gaussian inputs to evaluate convergence and performance. White Gaussian noise is compared with correlated inputs, which slow down the convergence. RLS is less affected by input correlation than LMS [1]. Performance metrics, including normalized misalignment (NM) and normalized projection misalignment (NPM) [9], assess the accuracy of estimated impulse responses.

3.7.1 The RLS-NLMS-T algorithm

Figures 3.1 and 3.2 show that the RLS$_M$-NLMS$_{N\text{-}M}$-T algorithm matches the convergence and tracking of the RLS-T algorithm but has slightly higher misalignment, similar to NLMS-T, especially for larger M. The RLS$_1$-NLMS$_3$-T and RLS-T algorithms perform similarly, suggesting the RLS update is best for the longest filter.

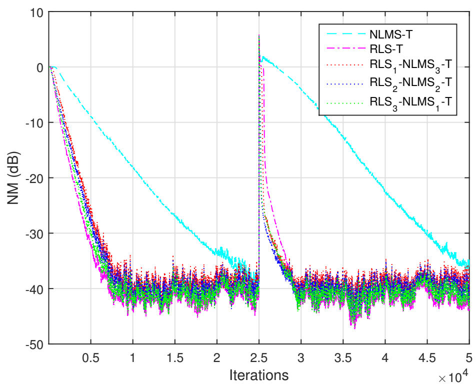

Fig. 3.1 NM of the NLMS-T, RLS-T, and RLS$_M$-NLMS$_{N\text{-}M}$-T algorithms in a tracking scenario, for different values of $M < N$

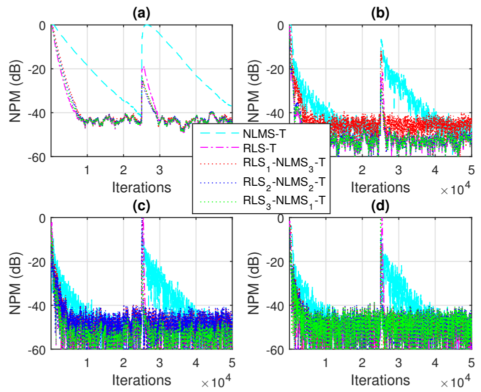

Fig. 3.2 NPM of NLMS-T, RLS-T, and RLS$_M$-NLMS$_{N\text{-}M}$-T, for different values of $M < N$, for (a) $\mathbf{h}_1$, (b) $\mathbf{h}_2$, (c) $\mathbf{h}_3$, (d) $\mathbf{h}_4$

3.7.2 The RLS-DCD-T algorithm

For $N = 4$ filters and $L = 2048$ ($L_1 = 16$, $L_2 = 8$, $L_3 = L_4 = 4$), tensor-based methods outperform RLS with AR(1) input (Figure 3.8), reducing RLS memory improves tracking but harms convergence. In low SNR with $L = 4096$ and $N = 4$ ($L_i = 8$), RLS-DCD-T maintains robustness like RLS-T (Figure 3.9).

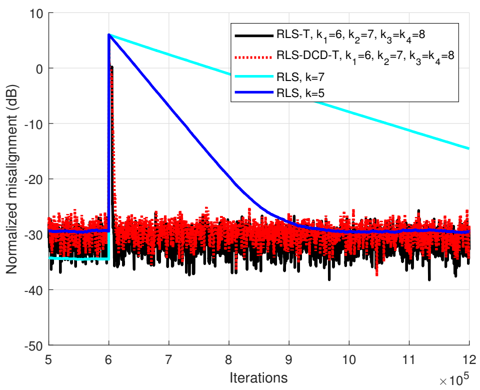

Fig. 3.8 Performance of the RLS-T, RLS-DCD-T, and RLS algorithms in a tracking scenario, for AR(1) sequence as input, with $N = 4$, and $L = 2048$

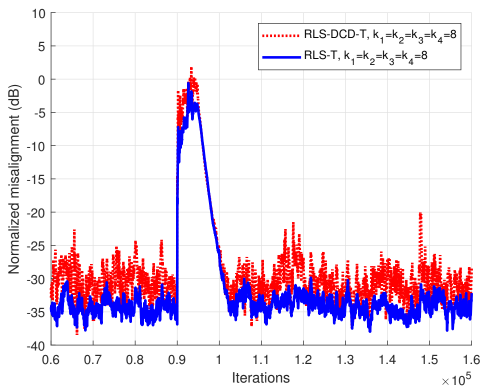

Fig. 3.9 Performance of the RLS-T and RLS-DCD-T algorithms, for AR(1) sequence as input, with $N = 4$, and $L = 4096$, SNR dropping to -15 dB for 5000 iterations

3.7.3 The VR-RLS-DCD-T algorithm

The VR-RLS-DCD-T algorithm, tested under low SNR with $N = 4$ systems, outperforms RLS in convergence speed after changes (Figure 3.11). It matches RLS-DCD-T and RLS-T performance with less computation efforts. Figure 3.13 shows VR-RLS-DCD-T's resilience to SNR drops, recovering faster than RLS variants.

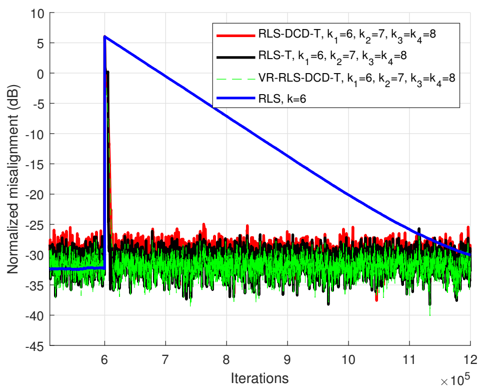

Fig. 3.11 Performance of the RLS-DCD-T, RLS-T, VR-RLS-DCD-T and RLS algorithms in a tracking scenario, for AR(1) sequence as input, with $N = 4$, and $L = 2048$

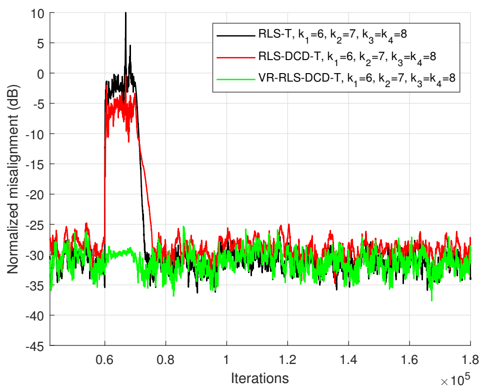

Fig. 3.13 Performance of the RLS-T, RLS-DCD-T, and VR-RLS-DCD-T algorithms, for AR(1) as input, with $N = 4$, and $L = 2048$, when SNR drops to -15 dB

3.8 Conclusions and perspectives

Adaptive algorithms in system identification face challenges in non-stationary and real-time environments. LMS methods offer a balance between performance and computational load, while RLS filters provide superior tracking and convergence but often face stability and complexity issues. Tensor-based approaches simplify high-dimensional tasks using shorter filters. The RLS_{M}-$\text{NLMS}_{\text{N-M}}$-T and RLS-DCD-T algorithms reduce complexity while retaining performance. VR-RLS-DCD-T excels in low SNR conditions, demonstrating robustness and efficiency even under high noise levels. Future work may focus on sliding window techniques and FPGA designs for MISO configurations, enhancing practical applications.

Chapter 4

Recursive least-squares adaptive algorithms based on data-reuse approach

This chapter enhances RLS adaptive algorithms with a data-reuse (DR) approach, enabling multiple updates on the same inputs to improve convergence and filter performance while reducing computational complexity. The DR-RLS and DR-RLS-DCD algorithms improve adaptive filtering and tracking, with the latter maintaining manageable complexity. The R-DR-RLS-DCD algorithm adds robustness in low signal-to-noise ratio (SNR) conditions. Simulations confirm the effectiveness of these methods, emphasizing the role of DR strategies in optimizing RLS algorithms for real-world applications. Results are published in [10–12].

4.1 The DR-RLS algorithm

Table 4.1 The DR-RLS algorithm

Step	**Actions**
Init.	Set $\widehat{\mathbf{h}}(0) = \mathbf{0}_{L\times 1},\ \mathbf{R}(0) = \delta \mathbf{I}_L,\ \delta > 0$ $\lambda = 1 - \frac{1}{KL},\ K \geq 1$
For $n = 1, 2, \ldots,$ number of iterations:	
1	Compute $\mathbf{x}(n)$
2	$\mathbf{k}(n) = \frac{\mathbf{R}^{-1}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^T(n)\mathbf{R}^{-1}(n-1)\mathbf{x}(n)}$
3	$e(n) = d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$
4	$\mathbf{R}^{-1}(n) = \frac{1}{\lambda}\left[\mathbf{I}_L - \mathbf{k}(n)\mathbf{x}^T(n)\right]\mathbf{R}^{-1}(n-1)$
5	$\alpha(n) = 1 - \mathbf{x}^T(n)\mathbf{k}(n)$
6	$\beta(n) = \frac{1-\alpha^{N_{\text{it}}}(n)}{1-\alpha(n)}$
7	$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \beta(n)\mathbf{k}(n)e(n)$

This section introduces the DR-RLS algorithm [13] (outlined in Table 4.1), which improves filter performance by allowing multiple updates on the same input and reference signals. This approach enhances convergence rates while managing computational complexity. The Kalman gain vector $\mathbf{k}(n)$ is used to update the inverse covariance matrix and filter coefficients, leveraging the data-reuse principle to perform N_{it} updates per data set. The algorithm efficiently consolidates updates, reducing computational cost to $O(L)$ operations, balancing tracking, accuracy, and misadjustment.

4.2 The DR-RLS-DCD algorithm

The DR-RLS-DCD algorithm [10, 11] integrates the data-reuse (DR) principle into the RLS-DCD adaptive algorithm to enhance tracking capabilities while slightly increasing arithmetic complexity. The algorithm applies multiple updates to the filter coefficients at each time index, using N_{it} iterations. The error signal and residual vector are adjusted iteratively, maintaining efficient computations. The DR-RLS-DCD algorithm's complexity remains proportional to the filter's length, offering faster convergence and improved tracking, crucial for real-time applications. The algorithm is resumed in Table 4.2, where

$$e_k(n) = \begin{cases} d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}(n) = e_0(n), & k = 0 \\ e_{k-1}(n) - \Delta\widehat{\mathbf{h}}_{k-1}^T(n)\mathbf{x}(n), & k \geq 1. \end{cases} \tag{4.18}$$

$$\mathbf{p}_k(n) = \begin{cases} \lambda \mathbf{r}_{N_{it}-1}(n-1) + \mathbf{r}_{e,x,0}(n), & k = 0 \\ \mathbf{r}_{k-1}(n) + \mathbf{r}_{e,x,k}(n), & k \geq 1, \end{cases} \tag{4.21}$$

Table 4.2 The exponential weighted DR-RLS-DCD algorithm

Step	**Actions**
Init.	Set $\widehat{\mathbf{h}}(0) = \mathbf{0}_{L\times 1},\ \mathbf{r}(0) = \mathbf{0}_{L\times 1}$ $\mathbf{R}(0) = \delta \mathbf{I}_L,\ \delta > 0$ $\lambda = 1 - \frac{1}{KL},\ K \geq 1$
For $n = 1, 2, \ldots,$ number of iterations :	
1	Compute $\mathbf{x}(n)$
2	Compute $\mathbf{R}^{(0)}(n) = \lambda \mathbf{R}^{(0)}(n-1) + \mathbf{x}(n)\mathbf{x}^{(0)}(n) \rightarrow \mathbf{R}(n)$
3	For $k = 0, 1, 2, \ldots, N_{it} - 1$:
3.a	Compute $e_k(n)$ using (4.18)
3.b	Compute $\mathbf{p}_k(n)$ using (4.21)
3.c	$\mathbf{R}(n)\Delta\mathbf{h}_k(n) = \mathbf{p}_k(n) \xrightarrow{\text{DCD}} \Delta\widehat{\mathbf{h}}_k(n), \mathbf{r}_k(n)$
4	$\widehat{\mathbf{h}}_k(n) = \widehat{\mathbf{h}}_k(n-1) + \Delta\widehat{\mathbf{h}}_k(n)$

4.3 The R-DR-RLS-DCD algorithm

The R-DR-RLS-DCD algorithm [12] integrates regularization and DR within the RLS-DCD framework to improve noise robustness, especially in low SNR conditions. The regularization parameter δ adjusts update frequency in the DCD component, with lower δ allowing more frequent updates and higher δ slowing them, which is advantageous in noisy environments [6, 14]. δ is determined using the adaptive filter's length, input signal variance, and SNR, ensuring the expected error aligns with the noise variance. This integration enhances convergence rates and robustness while maintaining computational efficiency, as outlined in Table 4.3.

Table 4.3 The R-DR-RLS-DCD algorithm

Init.	Set $\widehat{\mathbf{h}}(0) = \mathbf{0}_{L\times 1}$ $\mathbf{r}(0) = \mathbf{0}_{L\times 1}$ $\lambda = 1 - \frac{1}{KL},\ K \geq 1$
For $n = 1, 2, \ldots,$ number of iterations :	
1	Compute $\mathbf{x}(n)$
2	Compute $\mathbf{R}^{(0)}(n) = \lambda\mathbf{R}^{(0)}(n-1) + \mathbf{x}(n)\mathbf{x}^{(0)}(n) \longrightarrow \mathbf{R}(n)$
3	For $k = 1, 2, \ldots, N_{it} - 1$:
3.a	Compute $e_k(n)$ using (4.18)
3.b	Compute $\mathbf{p}_k(n)$ using (4.21)
3.c	$[\mathbf{R}(n) + \delta\mathbf{I}_L]\,\Delta\mathbf{h}_k(n) = \mathbf{p}_k(n) \xrightarrow{\text{DCD}} \Delta\widehat{\mathbf{h}}_k(n), \mathbf{r}_k(n)$
4	$\widehat{\mathbf{h}}_k(n) = \widehat{\mathbf{h}}_k(n-1) + \Delta\widehat{\mathbf{h}}_k(n)$

4.4 Simulation results

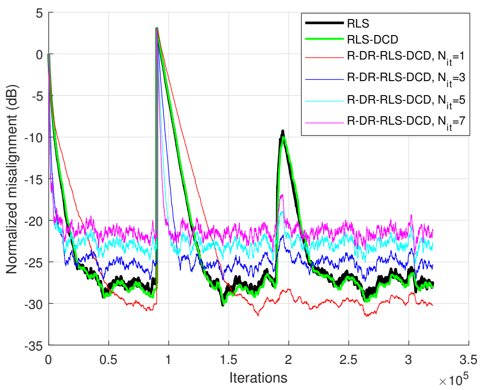

Fig. 4.5 Performance of the RLS, RLS-DCD, and R-DR-RLS-DCD algorithms for different values of N_{it} and fixed $N_u = 4$, for AR(1) sequence as input.

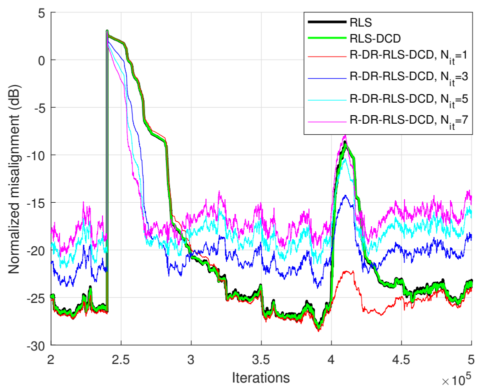

Fig. 4.6 Performance of the RLS, RLS-DCD, and R-DR-RLS-DCD algorithms for different values of N_{it}, for speech sequence as input.

Simulations for unknown system identification used the G.168 ITU-T recommendation with an impulse response length of $L = 128$. Two input signals, an AR(1) sequence and high-quality speech, were tested at an SNR of 10 dB. The forgetting factor was $\lambda = 1 - 1/(KL)$, with DCD parameters $N_u = 4$, $H = 1$, and $M_b = 16$.

The scenarios involved tracking (altering impulse response coefficients) and low SNR conditions, and compared R-DR-RLS-DCD, RLS, and RLS-DCD, using NM. For AR case, the changes for tracking occurred at index 90001, and starting with index 190001 the SNR dropped for 5000 iterations, as shown in Figure 4.5. For speech samples, the system changed at index 240001, with a 20 dB SNR drop between 400001 and 410000, as illustrated in Figure 4.6. Increasing N_{it} improved tracking but reduced steady-state accuracy. R-DR-RLS-DCD outperformed others in low SNR conditions. The results highlighted a trade-off between tracking speed and steady-state accuracy, with regularized algorithms showing less fluctuation under low SNR.

4.5 Conclusions and perspectives

This chapter explored advancements in low-complexity RLS algorithms, starting with the DR-RLS method and extending to DR-RLS-DCD and R-DR-RLS-DCD.

The DR-RLS-DCD and R-DR-RLS-DCD algorithms enhance RLS by incorporating the DCD technique, which efficiently solves auxiliary systems using simple arithmetic operations like additions and bit-shifts.

The DR-RLS algorithm lays the groundwork for subsequent developments, demonstrating the potential of data-reuse in adaptive filtering. Building on this, the DR-RLS-DCD algorithm introduces a novel data-reuse approach, balancing tracking speed, arithmetic complexity, and estimation accuracy. Its complexity is proportional to the filter length, making it suitable for hardware implementations. Simulations confirm its superior performance over methods based on Woodbury's identity, especially in noisy and highly correlated input scenarios.

The R-DR-RLS-DCD further improves capabilities by combining data-reuse and regularization techniques, enhancing tracking speed and robustness in low SNR conditions. Extensive testing shows its effectiveness across various input signals with increasing correlation levels, highlighting its promise in tracking scenarios with low SNR over extended iterations. Its hardware efficiency makes it ideal for applications requiring many filter coefficients.

Future directions include enhancing the DR-RLS-DCD with variable data-reuse iterations for improved accuracy and optimizing the R-DR-RLS-DCD for greater robustness in challenging environments. Exploring real-time applications of these algorithms is another significant opportunity.

Chapter 5

Recursive least-squares adaptive algorithms for stereophonic acoustic echo cancellation

This chapter explores SAEC systems with multiple loudspeakers and microphones. Traditional methods like LMS struggle with correlated inputs, while RLS algorithms are complex. The widely linear (WL) model improves performance in low SNR and double-talk (DT) scenarios using a single complex-valued filter.

WL-RLS-LSM algorithms and their variable regularized versions enhance tracking speed and accuracy. The WL-DR-RLS-DCD algorithm allows multiple updates per iteration for better tracking. Simulations confirm these methods' effectiveness, emphasizing the balance between complexity and performance. Results are published in [15–22].

5.1 Stereophonic acoustic echo cancellation based on the widely linear model

The SAEC setup uses a stereophonic configuration with two microphones and two loudspeakers per terminal to leverage binaural hearing. Each loudspeaker connects to a microphone via an unknown impulse response, creating echoes. The WL model simplifies echo cancellation by merging real-valued signals into a complex form:

$$x(n) = x_{\mathrm{L_c}}(n) + jx_{\mathrm{R_c}}(n) \tag{5.1}$$

This allows input signals to be encapsulated into a complex vector $\widetilde{\mathbf{x}}(n)$, using a single adaptive filter to estimate echo paths. The complex output is $y(n) = \widetilde{\mathbf{h}}_{\mathrm{t}}^{H}\widetilde{\mathbf{x}}(n)$, with $\widetilde{\mathbf{h}}_{\mathrm{t}}$ as the complex impulse response. The aposteriori error is $e(n) = d(n) - \widetilde{\mathbf{h}}^{H}(n)\widetilde{\mathbf{x}}(n)$. This reduces complexity and enhances filtering efficiency in SAEC.

5.2 The WL-RLS-LSM algorithms

The WL model effectively manages complex acoustic signal interactions in SAEC setups. The WL-RLS-LSM algorithms enhance this by using the exponentially weighted RLS method, where the forgetting factor λ balances tracking and accuracy. Recursive updates for the correlation matrix $\mathbf{R}(n)$ and cross-correlation vector $\mathbf{p}(n)$ ensure efficient computation, avoiding direct matrix inversion. These algorithms leverage the statistical properties of loudspeaker signals to solve auxiliary equations, combining RLS with line search methods (LSM) to maintain performance and stability, as detailed in Table 5.1. Initialization sets filter coefficients and residual vectors to zero, with $\mathbf{R}(n)$ initialized to avoid singularity. The algorithm iterates by updating the input vector:

$$\widetilde{\mathbf{x}}(n) = \left[x(n)\ x^*(n)\ \ldots\ x(n-L+1)\ x^*(n-L+1)\right]^T. \tag{5.11}$$

.

Table 5.1 The WL-RLS-LSM algorithm with complexity

Step	Actions	'×'
Init.	$\widetilde{\mathbf{h}}(0) = \mathbf{0}_{2L\times 1}$; $\mathbf{r}(0) = \mathbf{0}_{2L\times 1}$	0
	$\mathbf{R}(0) = \Phi\mathbf{I}_{2L},\ \Phi > 0$	0
For $n = 1, 2, \ldots,$ number of iterations:		
1	Update $\widetilde{\mathbf{x}}(n)$ using (5.1) and (5.11)	0
2	Update $\mathbf{R}(n)$ using time-shift property:	
	$\mathbf{R}_{:,1}(n) = \lambda\mathbf{R}_{:,1}(n-1) + x^*(n)\widetilde{\mathbf{x}}(n)$	$4L$
3	$\widetilde{y}(n) = \widetilde{\mathbf{h}}^H(n-1)\widetilde{\mathbf{x}}(n)$	$8L$
4	$e(n) = d(n) - \widetilde{y}(n)$	0
5	$\mathbf{p}_0(n) = \lambda\mathbf{r}(n-1) + e^*(n)\widetilde{\mathbf{x}}(n)$	$4L$
6	$\mathbf{R}(n)\Delta\mathbf{h}(n) = \mathbf{p}_0(n) \xrightarrow{\text{LSM}} \Delta\widetilde{\mathbf{h}}(n), \mathbf{r}(n)$	...
7	$\widetilde{\mathbf{h}}(n) = \widetilde{\mathbf{h}}(n-1) + \Delta\widetilde{\mathbf{h}}(n)$	0

5.3 The WL-RLS-LSM versions

The WL-RLS-LSM algorithms aim to solve an auxiliary system of equations by minimizing a cost function that represents the error between desired and estimated responses. The solution vector $\Delta\widetilde{\mathbf{h}}(n)$ is optimized by adjusting it iteratively to minimize this cost function, enhancing system response accuracy. This process utilizes input data statistics, starting with a zero vector and updating iteratively to converge to the optimal solution.

5.3.1 The conjugate gradient LSM

The conjugate gradient (CG) method, iteratively determines a direction vector $\mathbf{d}(k)$ and minimizes the cost function. Despite its complexity of $O(4L^2)$, it is popular for its performance and tracking speed.

5.3.2 The coordinate descent LSM

The coordinate descent (CD) method, exploits the statistical properties of the correlation matrix. It updates one component of the solution vector per iteration, achieving a complexity of $O(2LN_u)$.

5.3.3 The dichotomous coordinate descent LSM

The DCD method, uses a greedy approach and bit-shifts instead of multiplications, providing efficient updates. It is the most efficient method in terms of arithmetic operations, requiring only additions.

5.4 The VR-WL-RLS-LSM algorithms

This section introduces an enhancement to exponentially weighted RLS methods with a dynamic regularization parameter Φ, determined at each time index n based on the echo-to-noise ratio (ENR). This adjustment improves robustness in low ENR scenarios like double-talk (DT) by modifying the cost function to include Φ. The VR-WL-RLS-LSM algorithm implements this strategy, detailed in Table 5.5, which includes initialization and iterative updates involving real-valued divisions and a square root operation to compute Φ. Variances of signals $x(n)$, $d(n)$, and $\widetilde{y}(n)$, denoted as $c(n)$, are estimated using an exponential window to enhance performance in challenging acoustics:

$$\widetilde{\sigma}_c^2(n) = \gamma\widetilde{\sigma}_c^2(n-1) + (1-\gamma)\,|c(n)|^2\,. \tag{5.45}$$

Table 5.5 The VR-WL-RLS-LSM algorithm with complexity

Step	Actions	'×'	'+'
Init.	$\widetilde{\mathbf{h}}(0) = \mathbf{0}_{2L\times 1}$; $\quad \mathbf{r}(0) = \mathbf{0}_{2L\times 1}$	0	0
	$\mathbf{R}(0) = \widetilde{\Phi}(0)\mathbf{I}_{2L}$, $\widetilde{\Phi}(0) > 0$	0	0
	$\widetilde{\sigma}_x^2(n) = 0$; $\quad \widetilde{\sigma}_d^2(n) = 0$; $\quad \widetilde{\sigma}_{\widetilde{y}}^2(n) = 0$; $\quad \widetilde{\sigma}_{\widetilde{w}}^2(n) = 0$	0	0
For $n = 1, 2, \dots,$ number of iterations:			
1	Update $\widetilde{\mathbf{x}}(n)$ using (5.1) and (5.11)	0	0
2	Update $\mathbf{R}(n)$ using time-shift property: $\mathbf{R}_{:,1}(n) = \lambda\mathbf{R}_{:,1}(n-1) + x^*(n)\widetilde{\mathbf{x}}(n)$	$4L$	0
3	Update $\widetilde{\sigma}_{\widetilde{x}}^2(n)$, $\widetilde{\sigma}_d^2(n)$ using (5.45); $\quad \alpha = 2L\widetilde{\sigma}_{\widetilde{x}}^2(n)$	8	0
4	$\widetilde{y}(n) = \widetilde{\mathbf{h}}^H(n-1)\widetilde{\mathbf{x}}(n)$	$8L$	0
5	Update $\widetilde{\sigma}_{\widetilde{y}}^2(n)$ using (5.45)	4	0
6	$\widetilde{\sigma}_w^2(n) = \left\|\widetilde{\sigma}_d^2(n) - \widetilde{\sigma}_{\widetilde{y}}^2(n)\right\|$; $\quad \widetilde{\mathrm{ENR}} = \frac{\widetilde{\sigma}_{\widetilde{y}}^2(n)}{\widetilde{\sigma}_{\widetilde{w}}^2(n)}$	0	1
7	$\widetilde{\Phi}(n) = \left(1 + \sqrt{1 + \widetilde{\mathrm{ENR}}}\right)\alpha \,/\, \widetilde{\mathrm{ENR}}$	1	1
8	$e(n) = d(n) - \widetilde{y}(n)$	0	0
9	$\mathbf{p}_0(n) = \lambda\mathbf{r}(n-1) + e^*(n)\widetilde{\mathbf{x}}(n)$	$4L$	0
10	$\left[\mathbf{R}(n) + \widetilde{\Phi}(n)\mathbf{I}_{2L}\right]\Delta\mathbf{h}(n) = \mathbf{p}_0(n) \xrightarrow{\mathrm{LSM}} \Delta\widetilde{\mathbf{h}}(n), \mathbf{r}(n)$	...	...
11	$\widetilde{\mathbf{h}}(n) = \widetilde{\mathbf{h}}(n-1) + \Delta\widetilde{\mathbf{h}}(n)$	0	0

5.5 The WL-DR-RLS-DCD algorithm

In Section 4.2, the DR approach enhances RLS algorithms by updating filter coefficients $\widetilde{\mathbf{h}}(n)$ multiple times at the same index n using DCD. By looping steps 3-7 from Table 5.1 for N_{it} iterations, WL-DR-RLS-DCD aligns with WL-RLS-DCD when $N_{it} = 1$. Each iteration updates the filter estimate $\Delta\widetilde{\mathbf{h}}_q(n)$, error signal $e_q(n)$, and residual $\mathbf{p}_{0,q}(n)$.

The proposed algorithm, detailed in Table 5.6, maintains complexity proportional to the filter length times a small integer, with N_{it} generally less than 10 to balance tracking speed and computational effort.

Table 5.6 The WL-DR-RLS-DCD algorithm with complexity

Step	**Actions**	'×'	'+'
Init.	$\widetilde{\mathbf{h}}(0) = \mathbf{0}_{2L\times1}$; $\mathbf{r}(0) = \mathbf{0}_{2L\times1}$ $\mathbf{R}(0) = \Phi\mathbf{I}_{2L}$, $\Phi > 0$; $q = 0$	0 0	0 0
For $n = 1, 2, \ldots$:			
1	Update $\widetilde{\mathbf{x}}(n)$	0	0
2	Update $\mathbf{R}(n)$ using time-shift: $\mathbf{R}_{:,1}(n) = \lambda\mathbf{R}_{:,1}(n-1) + x^*(n)\widetilde{\mathbf{x}}(n)$	$4L$	$8L$
3	$q = q + 1$	0	0
4	Determine $e_q(n)$	$\leq 8L + 4(N_{it} - 1)N_u$	$\leq 6L - 2 + 2N_{it}$ $+ 4N_u(N_{it} - 1)$
5	Determine $\mathbf{p}_{0,q}(n)$	$4L(2N_{it} + 1)$	$4LN_{it}$
6	$\mathbf{R}(n)\Delta\mathbf{h}_q(n) = \mathbf{p}_{0,q}(n) \xrightarrow{\text{DCD}} \Delta\widetilde{\mathbf{h}}_q(n), \mathbf{r}_q(n)$	0	$\leq M_b + N_{it}[(8L+1)N_u]$
7	Determine $\widetilde{\mathbf{h}}_q(n)$	0	$\leq N_{it}N_u$
8	If $q < N_{it} \xrightarrow{\text{jump}}$ **Step 3**	0	0

5.6 Simulation results

In the SAEC setup, WL-RLS-based algorithms process input signals with diverse correlation properties, such as Gaussian noise and speech. A predistortion block reduces correlation between input channels, which can hinder performance, as detailed in [23]. Real-world acoustic impulse responses define the echo paths. Normalized misalignment was used to measure the performance.

5.6.1 The WL-RLS-LSM and VR-WL-RLS-LSM algorithms

The tracking and DT scenario was simulated with an ENR of 10 dB using a speech sequence as the input signal, predistorted with $\alpha_r = 0.35$. The unknown impulse responses had a length of $L = 256$, with $\lambda = 1 - 1/(64L)$ and $\gamma = 0.999$ for the VR algorithm. Echo paths changed at $t_0 = 120$, and a DT situation occurred from [230, 234] seconds, with performance measured using normalized misalignment. Figures 5.9, 5.10, and 5.11 show that VR algorithms demonstrated less performance reduction during DT occurrences, outperforming non-VR counterparts by over 7 dB, 4.5 dB, and 6 dB, respectively. Regularization improved accuracy at steady-state, and updates remained minimally affected by DT. Figure 5.12 indicates that VR versions exhibit similar misalignment curves, with VR-WL-RLS-DCD being the most attractive option as ENR decreases.

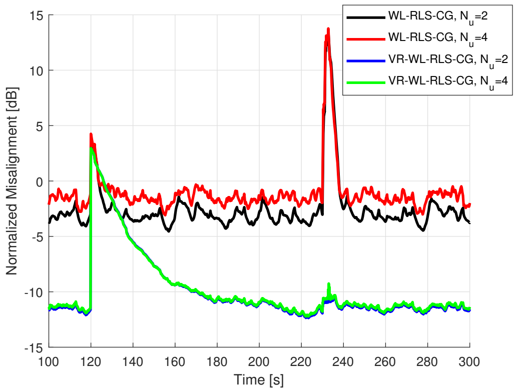

Fig. 5.9 Performance of the WL-RLS-CG and VR-WL-RLS-CG for different values of N_u, for speech sequence as input, with ENR set to 10 dB.

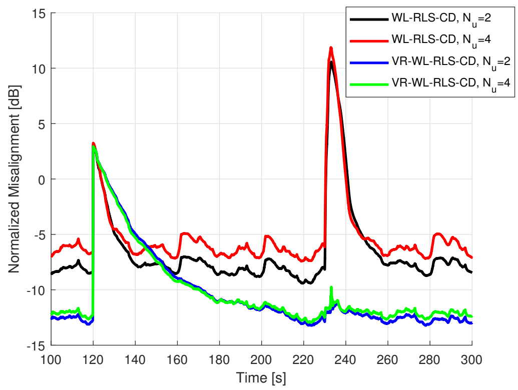

Fig. 5.10 Performance of the WL-RLS-CD and VR-WL-RLS-CD for different values of N_u, for speech sequence as input, with ENR set to 10 dB.

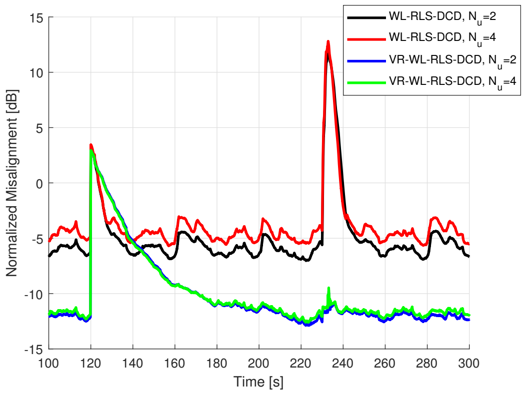

Fig. 5.11 Performance of the WL-RLS-DCD and VR-WL-RLS-DCD for different values of N_u, for speech sequence as input, with ENR set to 10 dB.

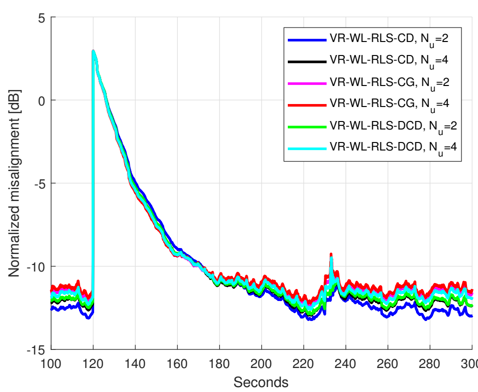

Fig. 5.12 Performance of the VR-WL-RLS-CG, VR-WL-RLS-CD, and VR-WL-RLS-DCD for different values of N_u, for speech sequence as input, with ENR set to 10 dB.

5.6.2 The WL-DR-RLS-DCD algorithm

The WL-DR-RLS-DCD algorithm's performance was validated for identifying the interleaved impulse response $\widetilde{\mathbf{h}}(n)$ using DCD parameters $H = 1$, $N_u = 4$, and $M_b = 16$ with SNRs of 25 dB and 30 dB.

Figure 5.16 shows that increasing N_{it} improved tracking speed, with $N_{it} = 3$ being sufficient for steady-state performance.

In Figure 5.17, swapping echo paths and microphone positions at steady-state improved tracking with higher N_{it} values up to a performance limit. The lack of preprocessing led to a significant performance gap between $\alpha_r = 0$ and $\alpha_r = 0.4$.

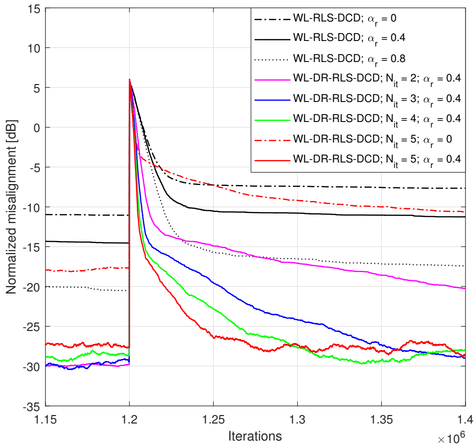

Fig. 5.16 Performance of the WL-RLS-DCD and WL-DR-RLS-DCD algorithms for different values of N_{it} and α_r. The input signal is an AR(1) sequence with the pole 0.9, and the length of the four unknown impulse responses is $L = 128$ and $\lambda = 1 - 1/(64L)$. The unknown system changes at time index 1200000.

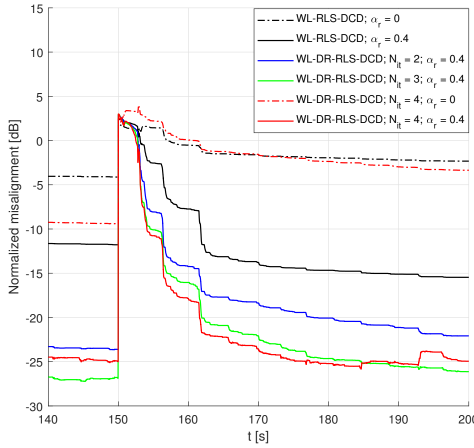

Fig. 5.17 Performance of the WL-RLS-DCD and WL-DR-RLS-DCD algorithms for different values of N_{it} and α_r. The input signal is a high-quality speech sequence, and the length of the four unknown impulse responses is $L = 256$ and $\lambda = 1 - 1/(96L)$. The unknown system changes at t = 150 s (microphones positions interchanged).

5.7 Conclusions and perspectives

This chapter explored low-complexity RLS algorithms for SAEC, emphasizing the WL framework's efficiency in handling complex-valued variables. Among the WL-RLS-LSM algorithms, the VR-WL-RLS-DCD is particularly efficient, using only additions and bit-shifts, making it ideal for practical applications. Simulations showed that these algorithms significantly reduce normalized misalignment by about 25 dB during DT intervals, with VR variants outperforming non-VR counterparts, especially at lower ENR levels.

The WL-DR-RLS-DCD algorithm enhances tracking while maintaining low complexity, adapting quickly to changes. It effectively improves tracking in scenarios with highly correlated inputs, despite a trade-off with steady-state accuracy. Overall, these algorithms balance tracking speed and steady-state performance, with the WL-DR-RLS-DCD standing out for hardware implementations in SAEC systems. Future work could focus on optimizing parameters and exploring multichannel extensions.

Chapter 6

Conclusions

This chapter summarizes the thesis's key findings, which advance adaptive filtering techniques with practical applications. The research tackled performance and complexity challenges by exploring various adaptive algorithms, enhancing both theoretical understanding and practical implementation. Significant improvements in tracking speed, convergence rates, and computational efficiency were demonstrated, enabling robust real-world applications. The thesis made novel contributions in three areas: enhancing system identification with tensor-based adaptive algorithms, improving efficiency with low-complexity RLS algorithms using a DR approach, and addressing SAEC challenges with RLS algorithms leveraging the widely linear model. These advancements collectively push the boundaries of adaptive filtering, offering new insights and solutions.

6.1 Obtained results

The research overviewed adaptive signal processing, highlighting its importance and applications, and defined the research scope and objectives.

Significant advancements were achieved in adaptive filtering, focusing on RLS and RLS-DCD algorithms. Traditional methods' limitations in nonstationary environments were addressed by introducing tensor-based adaptive algorithms, with RLS_{M}-$\text{NLMS}_{\text{N-M}}$-T achieving rapid convergence and reduced computational demands. RLS-DCD-T and VR-RLS-DCD-T maintained performance in low SNR conditions with minimal complexity. The DR-RLS method led to DR-RLS-DCD and R-DR-RLS-DCD, integrating the DCD technique for efficient solutions, balancing tracking speed, complexity, and accuracy, and excelling in noisy, correlated input environments.

For SAEC applications, the WL framework effectively managed complex-valued variables. LSM-based algorithms, particularly VR variants, showed notable improvements, with VR-WL-RLS-CD and VR-WL-RLS-DCD excelling in double-talk scenarios and maintaining tracking capabilities with minimal arithmetic workload, making them ideal for hardware implementations.

6.2 Original contributions

This section will provide a brief list of the original contributions. Each contribution will specify the original works from which it is derived, with references included in the following section.

[1, 2] Contributions to the development of an algorithm for MISO system identification that combines RLS and NLMS techniques, utilizing tensor-based methods.

[3, 4] Contributions to a new low-complexity algorithm for MISO system identification that integrates the exponentially weighted RLS algorithm with DCD iterations, utilizing tensor-based methods.

[5, 6] Contributions to the development of a variable-regularized version of the aforementioned low-complexity tensor-based algorithm for MISO system identification that integrates exponentially weighted RLS and DCD iterations, incorporating regularization terms within the cost functions to enhance performance.

[7, 8] Contributions to the development of a low-complexity RLS-based algorithm that utilizes a data-reuse methodology based on DCD iterations, enhancing tracking performance and convergence speeds in low signal-to-noise scenarios while minimizing costs for hardware implementations.

[9] Contributions to the development of a regularized version of the low-complexity RLS algorithm from [7, 8], which integrates the DCD method and employs a data-reuse methodology to enhance tracking speed and improve robustness in challenging SNR conditions.

[10] Contributions to the development of a decomposition-based approach for SAEC that utilizes the nearest Kronecker product, combining shorter filters formulated with the RLS algorithm, enhancing tracking speed and reducing computational complexity.

[11, 12, 13] Contributions to the integration of the complex-valued RLS algorithm with various line search methods in the SAEC framework, providing a theoretical analysis of the WL model and iterative techniques (CG, CD, and DCD).

[14, 15] Contributions to the development of a variable-regularized version of the above-mentioned family of adaptive LSM filters for double-talk scenarios, focusing on the adjustment of the correlation matrix to slow or freeze adaptation during disturbances.

[16, 17] Contributions to the integration of a data-reuse approach with the RLS algorithm and DCD iterations, enhancing computational efficiency and tracking performance in SAEC setups under highly correlated input conditions.

6.3 List of original publications

The following publications were supported by two grants from the Ministry of Research, Innovation and Digitization, CNCS–UEFISCDI: PN-III-P1-1.1-TE-2019-0529 and PN-III-P4-PCE-2021-0438, as part of PNCDI III. Additional support came from the National

Program for Research of the National Association of Technical Universities—GNAC ARUT 2023 (Grant 36/09.10.2023, code 133) and a grant from the Academy of Romanian Scientists (AOȘR) within AOȘR-TEAMS-III (project registration code 131/2024). Each publication received funding from different combinations of these grants.

[1] I.-D. Fîciu, C. Stanciu, C. Anghel, C. Paleologu, and L. Stanciu, "Combinations of Adaptive Filters within the Multilinear Forms," in *Proceedings of the 2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, Iași, Romania, pp. 1–4, 2021 **[BDI, IEEE Xplore]**.

[2] I.-D. Fîciu, L.-M. Dogariu, C.-L. Stanciu, and C. Paleologu, "Identification of Multilinear Forms Using Combinations of Adaptive Algorithms," in *Proceedings of the Sixth International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2021)*, Valencia, Spain, pp. 11–12, IARIA, 2021.

[3] I.-D. Fîciu, C.-L. Stanciu, C. Anghel, and C. Elisei-Iliescu, "Low-Complexity Recursive Least-Squares Adaptive Algorithm Based on Tensorial Forms," *Applied Sciences*, vol. 11, no. 18, 2021, WOS:000699216700001 **[ISI Q1, IF 2.5]**.

[4] I.-D. Fîciu, C.-L. Stanciu, L.-M. Dogariu, C. Elisei-Iliescu, and C. Paleologu, "Tensor-Based Recursive Least-Squares Algorithm with Low Computational Complexity," in *Proceedings of the Fifteenth International Conference on Sensor Technologies and Applications (SENSORCOMM 2021)*, Athens, Greece, pp. 32–33, IARIA, 2021.

[5] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, and C. Anghel, "Tensor-Based Recursive Least-Squares Adaptive Algorithms with Low-Complexity and High Robustness Features," *Electronics*, vol. 11, no. 2, 2022, WOS:000746966000001 **[ISI Q2, IF 2.6]**.

[6] I.-D. Fîciu, C. Elisei-Iliescu, C.-L. Stanciu, and C. Paleologu, "Variable-Regularized Low-Complexity RLS Adaptive Algorithms for Bilinear Forms," in *Proceedings of the Thirteenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2021)*, Porto, Portugal, pp. 9–12, IARIA, 2021.

[7] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, and R.-M. Udrea, "Low-Complexity Implementation of a Data-Reuse RLS Algorithm," in *Proceedings of the 2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, Prague, Czech Republic, pp. 289–293, 2022, WOS:001070846300060 **[ISI Proceedings]**.

[8] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, and C. Paleologu, "A Data-Reuse Approach for the RLS-DCD Algorithm," in *Proceedings of the 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications (IARIA Congress 2022)*, Nice, France, pp. 85–86, IARIA, 2022.

[9] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, C. Paleologu, and J. Benesty, "Low-Complexity Data-Reuse RLS Algorithm with Increased Robustness Features," in *Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies XI*, Constanța, Romania, vol. 12493, p. 124931V, International Society for Optics and Photonics, SPIE, 2023 **[BDI, Scopus]**.

[10] I.-D. Fîciu, C. Paleologu, J. Benesty, C. Elisei-Iliescu, C.-L. Stanciu, and C. Anghel, "A Decomposition-Based RLS Algorithm for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the 2022 International Symposium on Electronics and Telecommunications (ISETC)*, Timișoara, Romania, pp. 1–4, 2022 **[BDI, IEEE Xplore]**.

[11] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, L. Stanciu, and R.-M. Udrea, "Efficient RLS Algorithms with Line Search Methods for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the 2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, Iași, Romania, pp. 1–4, 2023 **[BDI, IEEE Xplore]**.

[12] I.-D. Fîciu and C.-L. Stanciu, "Stereophonic Acoustic Echo Cancellation Adaptive Algorithms using Line Search Methods," in *SD–ETTI 2023: 1st Doctoral Symposium on Electronics, Telecommunications, and Information Technology*, Bucharest, Romania, 2023.

[13] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, J. Benesty, C. Elisei-Iliescu, C. Anghel, and S. Ciochină, "Stereophonic Acoustic Echo Cancellation with the RLS Algorithm using the Conjugate Gradient Method," in *Advances in Signal Processing and Artificial Intelligence*, Tenerife (Canary Islands), Spain, pp. 55–57, IFSA Publishing, S. L., 2023.

[14] C.-L. Stanciu, C. Anghel, I.-D. Fîciu, C. Elisei-Iliescu, M.-R. Udrea, and L. Stanciu, "On the Regularization of Recursive Least-Squares Adaptive Algorithms Using Line Search Methods," *Electronics*, vol. 13, no. 8, 2024, WOS:001210704600001 **[ISI Q2, IF 2.6]**.

[15] C.-L. Stanciu, C. Anghel, C. Elisei-Iliescu, L.-M. Dogariu, I.-D. Fîciu, and C. Paleologu, "On the Regularization of a Low-Complexity Recursive Least-Squares Adaptive Algorithm," in *Proceedings of the Sixteenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2024)*, Venice, Italy, pp. 26–28, 2024.

[16] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, and J. Benesty, "Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation," *Applied Sciences*, vol. 13, no. 4, 2023, WOS:000938788600001 **[ISI Q1, IF 2.5]**.

[17] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, and F. Albu, "On the Performance of a Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the Eighteenth International Conference on Digital Telecommunications (ICDT 2023)*, Venice, Italy, pp. 6–9, IARIA, 2023.

6.4 Perspectives for further developments

This thesis highlights promising future research directions, including the exploration of sliding window techniques to improve adaptability in dynamic environments using tensor-based adaptive algorithms. Additionally, implementing fixed-point FPGA designs for MISO configurations could provide efficient solutions for real-world applications.

For low-complexity RLS algorithms using data-reuse strategies, enhancing the DR-RLS-DCD algorithm by varying the number of data-reuse iterations could improve steady-state accuracy. Optimizing the R-DR-RLS-DCD for greater robustness in challenging conditions, such as low SNR environments, is another promising avenue. Furthermore, implementing these algorithms in real-time systems may provide valuable insights and advancements.

For stereophonic acoustic echo cancellation, the WL-DR-RLS-DCD algorithm shows promise for hardware implementations. Future work might focus on optimizing these algorithms with variable parameters, like the number of data-reuse iterations and the forgetting factor, to improve performance in complex acoustic settings. Exploring impulse response decomposition with low-rank approximations and extending to multichannel scenarios could further enhance their applicability.

These perspectives underscore the potential for continued innovation, addressing both theoretical advancements and practical challenges in the field of adaptive filtering.

References

[1] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall information and system sciences series, Prentice Hall, 2002.

[2] I.-D. Fîciu, C. Stanciu, C. Anghel, C. Paleologu, and L. Stanciu, "Combinations of Adaptive Filters within the Multilinear Forms," in *Proceedings of the 2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, 2021.

[3] I.-D. Fîciu, L.-M. Dogariu, C.-L. Stanciu, and C. Paleologu, "Identification of Multilinear Forms Using Combinations of Adaptive Algorithms," in *Proceedings of the Sixth International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2021)*, pp. 11–12, IARIA, 2021.

[4] I.-D. Fîciu, C.-L. Stanciu, C. Anghel, and C. Elisei-Iliescu, "Low-Complexity Recursive Least-Squares Adaptive Algorithm Based on Tensorial Forms," *Applied Sciences*, vol. 11, no. 18, 2021.

[5] I.-D. Fîciu, C.-L. Stanciu, L.-M. Dogariu, C. Elisei-Iliescu, and C. Paleologu, "Tensor-Based Recursive Least-Squares Algorithm with Low Computational Complexity," in *Proceedings of the Fifteenth International Conference on Sensor Technologies and Applications (SENSORCOMM 2021)*, pp. 32–33, IARIA, 2021.

[6] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, and C. Anghel, "Tensor-Based Recursive Least-Squares Adaptive Algorithms with Low-Complexity and High Robustness Features," *Electronics*, vol. 11, no. 2, 2022.

[7] I.-D. Fîciu, C. Elisei-Iliescu, C.-L. Stanciu, and C. Paleologu, "Variable-Regularized Low-Complexity RLS Adaptive Algorithms for Bilinear Forms," in *Proceedings of the Thirteenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2021)*, pp. 9–12, IARIA, 2021.

[8] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[9] D. Morgan, J. Benesty, and M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Processing Letters*, vol. 5, no. 7, pp. 174–176, 1998.

[10] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, and R.-M. Udrea, "Low-Complexity Implementation of a Data-Reuse RLS Algorithm," in *Proceedings of the 2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 289–293, 2022.

[11] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, and C. Paleologu, "A Data-Reuse Approach for the RLS-DCD Algorithm," in *Proceedings of the 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications (IARIA Congress 2022)*, pp. 85–86, IARIA, 2022.

[12] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, C. Paleologu, and J. Benesty, "Low-Complexity Data-Reuse RLS Algorithm with Increased Robustness Features," in *Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies XI* (M. Vladescu, R. D. Tamas, and I. Cristea, eds.), vol. 12493, p. 124931V, International Society for Optics and Photonics, SPIE, 2023.

[13] C. Paleologu, J. Benesty, and S. Ciochină, "Data-Reuse Recursive Least-Squares Algorithms," *IEEE Signal Processing Letters*, vol. 29, pp. 752–756, 2022.

[14] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină, "Recursive Least-Squares Algorithms for the Identification of Low-Rank Systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 5, pp. 903–918, 2019.

[15] I.-D. Fîciu, C. Paleologu, J. Benesty, C. Elisei-Iliescu, C.-L. Stanciu, and C. Anghel, "A Decomposition-Based RLS Algorithm for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the 2022 International Symposium on Electronics and Telecommunications (ISETC)*, pp. 1–4, 2022.

[16] I.-D. Fîciu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, L. Stanciu, and R.-M. Udrea, "Efficient RLS Algorithms with Line Search Methods for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the 2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, 2023.

[17] I.-D. Fîciu and C.-L. Stanciu, "Stereophonic Acoustic Echo Cancellation Adaptive Algorithms using Line Search Methods," in *SD–ETTI 2023: 1st Doctoral Symposium on Electronics, Telecommunications, and Information Technology*, 2023.

[18] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, J. Benesty, C. Elisei-Iliescu, C. Anghel, and S. Ciochină, "Stereophonic Acoustic Echo Cancellation with the RLS Algorithm using the Conjugate Gradient Method," in *Advances in Signal Processing and Artificial Intelligence* (S. Y. Y. Prof., Dr., ed.), pp. 55–57, IFSA Publishing, S. L., 2023.

[19] C.-L. Stanciu, C. Anghel, I.-D. Fîciu, C. Elisei-Iliescu, M.-R. Udrea, and L. Stanciu, "On the Regularization of Recursive Least-Squares Adaptive Algorithms Using Line Search Methods," *Electronics*, vol. 13, no. 8, 2024.

[20] C.-L. Stanciu, C. Anghel, C. Elisei-Iliescu, L.-M. Dogariu, I.-D. Fîciu, and C. Paleologu, "On the Regularization of a Low-Complexity Recursive Least-Squares Adaptive Algorithm," in *Proceedings of the Sixteenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2024)*, pp. 26–28, 2024.

[21] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, and J. Benesty, "Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation," *Applied Sciences*, vol. 13, no. 4, 2023.

[22] I.-D. Fîciu, C.-L. Stanciu, C. Paleologu, and F. Albu, "On the Performance of a Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation," in *Proceedings of the Eighteenth International Conference on Digital Telecommunications (ICDT 2023)*, pp. 6–9, IARIA, 2023.

[23] J. Benesty, C. Paleologu, T. Gaensler, and S. Ciochină, *A Perspective on Stereophonic Acoustic Echo Cancellation*, vol. 4. Springer, 01 2011.